Sbs Commands

sbs-backup

Use the **sbs-backup** command to back up and restore the configuration of the Cisco Cyber Vision Center.

sbs-backup *command* [*args..*]

Syntax Description

export	Creates a backup of the Cisco Cyber Vision Center. The backup includes the configuration, data, and sensor management information.
import *file*	Restores a Cisco Cyber Vision Center from the specified backup archive. **Note** All existing data and configuration on the Center will be erased and replaced with the data from the backup.

Command History

Release	Modification
4.0.0	This command was introduced.

This example displays how to back up Cisco Cyber Vision Center data:

```
root@center100:~# sbs-backup export
Usbs-backup export
Please note that license information is also backed up and will be restored if you restore
 the backup on the same system from which the backup was taken.
If you restore the backup on a different system, first return the license reservation to
Cisco Smart Software Licensing so you can set it up again after the restoration on the new
 system.
***************** Taking backup of file system     *****************
***************** Taking backup of database        *****************
***************** Taking backup of RMQ definitions *****************
***************** Taking backup of center version  *****************
***************** Taking backup of symlinks        *****************
***************** Taking backup of extension       *****************
Created center archive at
/data/tmp/ccv-center-backup/ccv-center-backup-Center-5.0.1-20240927153623.tar.gz
```

This example displays how to restore a configuration from an archive file:

```
root@center100:~# sbs-backup import
/data/tmp/ccv-center-backup/ccv-center-backup-Center-5.0.1-20240927153623.tar.gz
Usbs-backup import
/data/tmp/ccv-center-backup/ccv-center-backup-Center-5.0.1-20240927153623.tar.gz
***************** Restoring file system     *****************
***************** Restoring database        *****************
***************** Restoring RMQ definitions *****************
***************** Restoring symlinks        *****************
***************** Restoring extension       *****************
Restore completed, please reboot to finalise the system configuration. After reboot, please
 install the Reports extension compatible with the center version.
```

sbs-closest-sensor-mode

Use the **sbs-closest-sensor-mode** to control the display of sensor and PCAP data sources associated with assets. By default, the **sbs-closest-sensor-mode** option is disabled.

sbs-closest-sensor-mode *action*

Syntax Description

enable	Displays the **Seen By** column in the **Assets seen in current active view** under the **Asset Visiblity** section of the Cisco Cyber Vision User Interface.
disable	Displays the **Data Sources** column in the **Assets seen in current active view** under the **Asset Visiblity** section of the Cisco Cyber Vision User Interface.

Command History

Release	Modification
5.2.0	This command was introduced.

To enable the closest sensor mode, run this command:

```
root@center100:~# sbs-closest-sensor-mode enable


Enabling closest sensor mode...
Successfully enabled!
```

To disable the mode, run this command:

```
root@center100:~# sbs-closest-sensor-mode disable


Disabling closest sensor mode...
Successfully disabled!
```

sbs-db

Use the **sbs-db** command to manage and interact with a database. It provides a wide range of functionalities for database administration, maintenance, data manipulation, and troubleshooting.

sbs-update *commands* [*args..*]

Syntax Description

`aggregate-flows`	Enables or disables the aggregation of flows based on client port.
`cleanup`	Cleans up the database.
`connect`	Opens a psql shell (PostgreSQL interactive terminal).
`count`	Counts rows in all tables.
`count-short`	Counts rows in relevant tables.
`create-extensions`	Creates database extensions like `hstore` and `pg_stat_statements`
`destroy`	Drops the database and all its data.
`drop-matviews`	Deletes all warehouse materialized views.
`dump`	Dumps all database content. **Note** By default, the database dump is stored in the `/data/tmp` folder. The filename is i `sbs-data-dump-<FQDN of the center>-<centertype>-<center version>-<timestamp>..sql.gz`. For example, `sbs-db-dump-centerdoc165labautomccvlocal-standalone-5.2.0`
`dump-tables`	Dumps specific tables.
`execute`	Executes a SQL query.
`exec-pretty`	Executes a SQL query and formats the output for readability.
`find-schema-files`	Lists the SQL files that are loaded during initialization.
`force-expiration`	Forces immediate data expiration.
`import-snort`	Imports Snort rules and categories
`indexes-size`	Displays the size of all indexes.
`init`	Creates the database user and database.
`init-load`	Creates the database user and database, and loads data from a provided file.
`list-custom-networks`	Lists all custom networks.
`list-extensions`	Lists all installed extensions.

`list-migrations`	Lists database migrations by time.
`list-schemas`	Lists all database schemas.
`list-storage-settings`	Lists all storage settings.
`list-tables`	Displays table names by schema.
`list-triggers`	Displays triggers by schema.
`load`	Loads an SQL command file into the database.
`migrate`	Migrates the database.
`optimize`	Optimizes the database using `VACUUM` if a flag is defined.
`port-scan-detection`	Enables or disables port scan detection.
`purge-components`	Removes components and associated data.
`purge-credentials-until`	Removes all credentials until a specified date.
`purge-events`	Removes events between dates with a specific metadata ID.
`purge-events-since`	Removes all events since a specified date.
`purge-events-until`	Removes all events until a specified date.
`purge-external-communications`	Removes components and associated data.
`purge-flows`	Removes all flows by tag.
`purge-flows-since`	Removes all flows since a specified date.
`purge-flows-until`	Removes all flows until a specified date.
`purge-orphan-components`	Removes all orphan components.
`purge-since`	Removes flows, events, and variables since a specified date.
`purge-until`	Removes flows, events, and variables until a specified date.
`purge-variables-since`	Removes all variables since a specified date.
`purge-variables-until`	Removes all variables until a specified date.
`remote-access-protocol`	Adds or removes remote access protocols.
`remote-domain-regex`	Adds or removes remote access domain regular expressions.
`reset-data`	Repacks tables.
`reset-group-impact`	Updates group criticality.
`reset-password`	Resets a user's password.
`reset-users`	Removes all users.

`restore`	Restores a database dump.

Command History

Release	Modification
4.0.0	This command was introduced.

This example displays how to dump all the content of a database:

```
root@center100:~# sbs-db dump
dump database to:
/data/tmp/sbs-db-dump-centerdoc165labautomccvlocal-standalone-5.2.0-20250423042226.sql.gz
```

This example displays how to reset the password of a database user:

```
root@center100:~# sbs-db reset-password user@cisco.com
User password successfully reset. You can now set a new password by login to the GUI using
 this temporary password: iO********l26fqc
```

This example displays how to connect to the database using psql:

```
root@center100:~# sbs-db connect
```

This example displays how to purge all flows since a specific date:

```
root@center100:~# sbs-db purge-flows-since "2024-01-01"
```

sbs-device-engine

Use the **sbs-device-engine** command to group various components within a system into logical devices. It interacts with a database and uses a configuration file to control its behavior.

sbs-device-engine [*options*]

Syntax Description

`-center-id string`	Overrides the default Center ID. `string`: The Center ID to use.
`-db_host string`	Specifies the database hostname. `string`: The hostname or IP address of the database server.
`-db_name string`	Specifies the database name. `string`: The name of the database.
`-db_password string`	Specifies the database user password. `string`: The password for the database user.
`-db_port int`	Specifies the database port. `int`: The port number on which the database server is listening.
`db_user string`	Specifies the database username. `string`: The username for accessing the database.
`-f string`	Specifies the configuration filename. Default: `"/data/etc/sbs/device-engine.conf"` `string`: The path to the configuration file
`-loglevel string`	Specifies the logging verbosity level. `string`: The logging level (for example, debug, info, warning, error).
`-logoutput string`	Specifies the logging output. `string`: The location where logs should be written (for example, a file path or "stdout").

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to group components into devices:

```
root@center100:~# sbs-device-engine
06/06/2024 12:32:48 +0000 undefined INFO CenterID provided by: /data/etc/sbs/center-id
```

```
                            caller=config.go:236
06/06/2024 12:32:48 +0000 undefined INFO center ID: 3ea90f42-3830-ac99-b12e-efd0c64fda7d
                            caller=config.go:261
06/06/2024 12:32:48 +0000 device-engine INFO Center type: standalone
                          caller=postgres.go:587
06/06/2024 12:32:48 +0000 device-engine INFO Connected to postgres on /var/run/postgresql:5432
 with user: ics on dbname: ics caller=app.go:35
06/06/2024 12:32:48 +0000 device-engine INFO RabbitMQ available
                         caller=connection.go:29
06/06/2024 12:32:48 +0000 device-engine INFO Using default activity tags period: 1 month
                            caller=engine.go:144
06/06/2024 12:32:48 +0000 device-engine INFO Number of components taken into account: 13
                            caller=engine.go:135
06/06/2024 12:32:48 +0000 device-engine INFO Rule SwitchAggregation is disabled
                             caller=rules.go:179
06/06/2024 12:32:48 +0000 device-engine INFO Creating 6 devices covering 10 components
                       caller=save_devices.go:31
06/06/2024 12:32:48 +0000 device-engine INFO exiting...
                               caller=main.go:42
```

To use a custom configuration file:

```
root@center100:~# sbs-device-engine -f /path/to/myconfig.conf
```

sbs-diag

Use the **sbs-diag** command to extract the diagnostic files from the Cisco Cyber Vision Center.

The diagnostic file will be copied to the `/data/tmp` folder with the `sbs-diag-export-<CENTERTYPE>-<CENTERNAME>-<DATETIME>.tgz` name.

sbs-diag [**OPTIONS**]

Syntax Description

-h	To generate the command help.
-o	To specific the path where the generated diagnostics are saved. If you do not specify the path, the file will be copied to the `/data/tmp` folder.
-v	To use the verbose mode.
-n	To generate diagnostics without accessing the database.
-l	To generate a reduced version of the diagnostics.
-b	To generate additional benchmarking by adding more information such as disk performances in the diagnostics. **Note** This is a CPU-intensive activity, and several processes including the user interface will be unavailable for a few minutes during benchmarking.

Command History

Release	Modification
3.0	This command was introduced.

This example displays how to change the network configuration,:

```
root@center100:~# sbs-diag
[2024-06-06 12:43:10.550568354] [sbs-diag:126221] Exporting diagnostics data...
[2024-06-06 12:43:10.563196830] [sbs-diag:126221] - Gathering system data
[2024-06-06 12:43:10.574734602] [sbs-diag:126221] - Gathering hardware data
[2024-06-06 12:43:19.307577251] [sbs-diag:126221] - Journal error+warning
[…]
[2024-06-06 12:43:21.266439032] [sbs-diag:126221] - Configs
[…]
[2024-06-06 12:43:26.591608618] [sbs-diag:126221] - pg_stats
[…]
[2024-06-06 12:43:35.061380378] [sbs-diag:126221] - Compressing data
[2024-06-06 12:43:36.172173931] [sbs-diag:126221] - Deleting temporary data
[2024-06-06 12:43:36.192001612] [sbs-diag:126221] Archive
/data/tmp/sbs-diag-export-standalone-center100-202406061243.tgz is ready.
```

sbs-system-fqdn

Use the **sbs-system-fqdn** command to check the fully qualified domain name (FQDN) of the Cisco Cyber Vision Center.

sbs-system-fqdn

Command History

Release	Modification
Release 4.0.0	This command was introduced.

This example displays how to check your center's FQDN:

```
Device# /opt/sbs/bin/sbs-system-fqdn
center100.sentryo.local
```

sbs-erase

Use the **sbs-erase** command to factory reset the Cisco Cyber Vision Center configuration.

Note Running this command will remove all data from the system.

sbs-erase

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to reset the Center configuration:

```
root@center100:~# sbs-erase
This will reboot and destroy all data on this system. Type yes in capital letters to confirm:
 YES
Connection to 10.2.3.100 closed by remote host.
```

sbs-netconf

Use the **sbs-netconf** command to reconfigure or add network routes to your Cisco Cyber Vision Center's Ethernet interfaces.

sbs-netconf

Command History

Release	Modification
4.0.0	This command was introduced.

This example displays how to change the network configuration,:

```
root@center100:~# sbs-netconf
```

On running this command, a configuration window opens. You can follow the options there.

sbs-passwd

Use the **sbs-passwd** command to change the password of the cv-admin user of the Cisco Cyber Vision Center.

Note After the fresh installation, you must change the cv-admin password.

sbs-passwd

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to change the password:

```
root@center100:~# sbs-passwd
Password must be at least 16 characters long.
Password must contain characters from at least 3 of the following characters class:
    lowercase, capitals, numbers or punctuation.
Enter password:
Confirm Password:
```

sbs-system-fqdn

Use the **sbs-system-fqdn** command to check the fully qualified domain name (FQDN) of the Cisco Cyber Vision Center.

sbs-system-fqdn

Command History

Release	Modification
Release 4.0.0	This command was introduced.

This example displays how to check your center's FQDN:

```
Device# /opt/sbs/bin/sbs-system-fqdn
center100.sentryo.local
```

sbs-timeconf

Use the **sbs-timeconf** command to change the Network Time Protocol (NTP) parameters.

sbs-timeconf [**-h**] [**-a** *sensor_serial_number*] [**-r** *sensor_serial_number*] [**-p** *dest_directory_path center_ip_address sensor_serial_number*] [**-g**] [**-n** *network_address network_mask*] [**-m** *network_address network_mask*] [**-s** *server_ip_address* [*key_id AES128CMAC_key_value...*]] [**-t** *server_ip_address*]

Syntax Description

-a *sensor_serial_number*	Adds a sensor with the specified serial number to the configuration. This allows the system to communicate with that sensor.
-r *sensor_serial_number*	Removes a sensor with the specified serial number from the configuration. This stops the system from communicating with that sensor.
-p *dest_directory_path center_ip_address sensor_serial_number*	Generates provisioning files for a given sensor. These files likely contain configuration data that is needed for the sensor to operate.
-g	Generates base configuration files for the `ntpd` daemon, which is the NTP daemon that are used for time synchronization.
-n *network_address network_mask*	Allows machines on the specified network to communicate with the Cisco Cyber Vision Center.
-m *network_address network_mask*	Revokes communication access for machines on the specified network.
-s *server_ip_address* [*key_id AES128CMAC_key_value ...*]	Adds an NTP server with the specified IP address to the configuration. It also supports authentication using AES128CMAC with a key ID and key value.
-t *server_ip_address*	Removes an NTP server with the specified IP address from the configuration.

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to add an NTP parameter:

```
root@center100:~# sbs-timeconf -s time1.google.com
```

This example displays how to remove an NTP parameter:

```
root@center100:~# sbs-timeconf -t time1.google.com
```

sbs-update

Use the **sbs-update** command to perform various operations related to software update of the Cisco Cyber Vision Center application including integrity checks, upgrade, and rollback.

sbs-update *commands* [*options*]

Syntax Description

check *file*	Checks the integrity of the specified update file.
prepare-install *file* *dir* [**allow-rollback**]	Performs the integrity check and prepares the update for installation after the next reboot. The **allow-rollback** flag enables you to roll back the update if needed.
install *file*	Installs the update from the specified file. The update is only installed if the version is newer than the currently installed version and has a valid signature.
install-with-rollback *file*	Installs the update and allows rollbacks. This means you can revert to the previous version if needed.
-undo *network_address* *network_mask*	Reverts to the last installed update.
update-script *file* *dir* [**allow-rollback**]	Updates the `sbs-update.sh` script itself. The **allow-rollback** flag enables you to roll back the update if needed.

Command History

Release	Modification
3.0.0	This command was introduced.

This example displays how to upgrade the Cisco Cyber Vision Center:

```
root@center100:~# sbs-update install /data/tmp/CiscoCyberVision-update-center-5.0.0.dat
Extracted archive directory /data/tmp/sbs-update.Mzb0PA/files
Installed version 4.4.0+202405071704
Updated version 5.0.0+202405241346
Preparing /data/tmp/CiscoCyberVision-update-center-5.0.0.dat for setup on next reboot.
WARNING: rebooting...
```